



# **Automated Payments and Merchant Query Interfaces Guide**

**For use by all Merchants**

This guide describes how to connect to the Automated Payments and Merchant Query Interfaces.

**[www.skrill.com](http://www.skrill.com)**

Version 3.0

## Copyright

© 2021. Skrill Ltd. All rights reserved.

The material contained in this guide is copyrighted and owned by Skrill Ltd together with any other intellectual property in such material. Except for personal and non-commercial use, no part of this guide may be copied, republished, performed in public, broadcast, uploaded, transmitted, distributed, modified or dealt with in any manner at all, without the prior written permission of Skrill Ltd, and, then, only in such a way that the source and intellectual property rights are acknowledged.

To the maximum extent permitted by law, Skrill Ltd shall not be liable to any person or organisation, in any manner whatsoever from the use, construction or interpretation of, or the reliance upon, all or any of the information or materials contained in this guide.

The information in these materials is subject to change without notice and Skrill Ltd. assumes no responsibility for any errors.

***Skrill Ltd.***

Registered office: Skrill Limited, 25 Canada Square, Canary Wharf, London, E14 5LQ, UK.

## Version Control

Date	Version	Description
April 2015	2.3	Removal of Slovakian Koruna, Estonian Kroon, and Lithuanian Litas from supported currencies.
July 2015	2.4	Changed URLs from <a href="http://www.moneybookers.com">www.moneybookers.com</a> to <a href="http://www.skrill.com">www.skrill.com</a> . Changed all methods incorrectly listed as POST to GET.
September 2015	2.5	Listed the ports that can be used with the <code>refund_status_url</code> . Corrected a number of instances in the refunds section where the <code>md5sig</code> examples were shown in lower rather than upper case. Clarified the explanation of the amount field used to prepare a partial refund. Corrected the description of the secret word.
September 2016	2.6	Updated examples illustrating use of escape sequences. Test account details added.
July 2017	2.7	Added details of Quick Checkout payouts. Added a missing error and status value.
November 2017	2.8	Updated Quick Checkout Payouts chapter.
February 2018	2.8	Added <code>COUNTRY_BLOCKED_FOR_PAYOUTS</code> error.
March 2018	2.8	Added Colombian Peso as supported currency.
October 2018	2.8	Modified Quick Checkout payout restrictions.
January 2019	2.8	Modified the Security Restrictions to include the steps to enable the Quick Checkout Secure Restriction service. Modified Page Numbering.
April 2019	2.8	Added <code>account_id</code> parameter Added specifications for extended currency account information
April 2021	2.9	Added account balances section in Get account balances page.
July 2021	3.0	Modified the Contact for Queries section in Introduction. Modified the Separate API/MQI Password section in Security.

Publication number: *GW-API-REL-3/28/19*

# Contents

<b>1. ABOUT THIS GUIDE .....</b>	<b>1</b>
1.1. Related documentation .....	1
1.2. Conventions used in this guide .....	1
<b>2. INTRODUCTION .....</b>	<b>2</b>
2.1. Merchant test account.....	3
2.2. Separating balances by fund origin.....	3
2.3. Contact for queries .....	3
<b>3. SECURITY .....</b>	<b>4</b>
3.1. Security requirements.....	4
3.1.1. Accept request-header field .....	4
3.1.2. Content types.....	5
3.2. Separate API/MQI password.....	5
<b>4. SEND MONEY USING AN HTTPS REQUEST .....</b>	<b>7</b>
4.1. Sending a transfer prepare request .....	7
4.2. Executing a transfer request .....	9
4.3. Reposting a transfer request.....	11
<b>5. QUICK CHECKOUT PAYOUTS .....</b>	<b>12</b>
5.1. Sending a payout prepare request.....	12
5.2. Executing a payout transfer request.....	14
5.3. Payout status report .....	16
5.3.1. MD5 signature .....	17
<b>6. REFUNDS .....</b>	<b>18</b>
6.1. Preparing a refund .....	18
6.2. Executing a refund .....	21
6.3. Refund status report .....	22
<b>7. MERCHANT QUERY INTERFACE .....</b>	<b>24</b>
7.1. MQI Actions .....	24
7.1.1. Repost transaction status .....	25
7.1.2. View transaction status.....	25
7.1.3. View account history .....	27
7.1.4. Cancel a recurring payment.....	28
7.1.5. View recurring payment status.....	28
7.1.6. Extend the end date of a recurring payment.....	29
7.1.7. Cancel a Skrill 1-Tap payment.....	29
7.1.8. View Skrill 1-Tap payment status.....	30
7.1.9. Get account balances.....	30
7.2. Error messages .....	31
<b>8. APPENDICES .....</b>	<b>32</b>
8.1. ISO 4217 currencies .....	32
8.2. Error Messages.....	33
<b>9. GLOSSARY.....</b>	<b>35</b>
<b>INDEX .....</b>	<b>37</b>

---

## 1. ABOUT THIS GUIDE

---

This guide supplements the Quick Checkout or Wallet Checkout integration guide (as appropriate) and describes how to use the Skrill Automated Payments Interface (API) and Skrill Merchant Query Interface (MQI), which provide alternative means of connecting to the Skrill Payment Gateway and can be used to automate transaction requests such as transferring money to another Skrill account (used for mass payment), queries and refunds (where available - this feature is not supported for all merchant types).

The guide covers the different transaction request options and describes how to implement them.

You will need a working knowledge of HTTP(S) Requests and XML and an understanding of how to send and receive information using these protocols.

### 1.1. Related documentation

You should use this guide together with the Skrill documents described below. If you are a new Skrill merchant, read one of these guides first.

**Table 1-1: Other Guides**

Guide	Description
<i><b>Skrill Quick Checkout Integration Guide</b></i>	Describes how to integrate and customise the Skrill Quick Checkout. Applicable to Ecommerce merchants.
<i><b>Skrill Wallet Checkout Integration Guide</b></i>	Describes how to integrate and customise the Skrill Wallet Checkout and 1-Tap payment method. Applicable to Wallet merchants.

### 1.2. Conventions used in this guide

The table below lists some of the conventions used in this guide.

**Table 1-2: List of conventions**

Convention	Description
<i><b>Reference</b></i>	Indicates a reference to another section in this guide. For example, refer to <i><b>User Administration on page 34.</b></i>
<code>Code example</code>	Used to illustrate example code, functions and commands.
<i><b>File path</b></i>	Used to indicate a file path or folder structure.
<u>Glossary</u>	Glossary term
Menu1 > Menu option2 >	Indicates a menu path.

---

## 2. INTRODUCTION

---

The Skrill Automated Payments Interface (API) and Merchant Query Interface (MQI) enable you to execute automated requests to Skrill.

The API is used for the following functions:

- Refund Quick Checkout / Wallet Checkout / 1-Tap payments. (where available)
- Transfer Money to another Skrill Account (send money)
- Taking subsequent 1-Tap payments (after the initial setup payment) - Wallet Checkout merchants only
- Pay out funds (such as winnings) to customers by referencing an original QCO transaction (payment) made on the gateway

The MQI is used for the following functions:

- Repost transaction status information for payment transactions (Wallet/Quick checkout payments and 1-Tap subsequent payments)
- View transaction status (payment and send money transactions)
- Download and view account history
- Cancel a recurring payment - Wallet Checkout merchants only
- View the status of a recurring payment - Wallet Checkout merchants only
- Extend the end date of a recurring payment - Wallet Checkout merchants only
- Cancel a 1-Tap payment - Wallet Checkout merchants only
- View the status of a 1-Tap payment - Wallet Checkout merchants only

The API uses the following URLs:

- <https://www.skrill.com/app/pay.pl>
- <https://www.skrill.com/app/refund.pl>

The MQI uses the URL: <https://www.skrill.com/app/query.pl>

---

**Note:** We strongly advise that you call the Skrill URLs by hostname when making requests rather than hard-coding the static IP of the Skrill server, which is subject to change.

---

## 2.1. Merchant test account

Skrill recommends that you open a merchant test account to help you become familiar with the Automated Payments Interface. Test accounts operate in the live environment, but funds cannot be sent from a test account to a live account.

To obtain a test account, please register a personal account at [www.skrill.com](https://www.skrill.com), and then contact the [Merchant Services](#) team with the account details so that they can enable it.

---

**Note:** When testing payments, you should set the beneficiary email to the same account in order to ensure that the test account has sufficient virtual funds.

---

## 2.2. Separating balances by fund origin

To comply with licensing agreements, merchant account balance funds must be separated by origin into one of two groups. The groups are as follows:

- European Economic Area
- Non-European Economic Area

In most cases, Skrill can automatically distribute the funds based on the country of residence of the sender (for deposits) or the recipient (for send money transactions). When Skrill is unable to automatically separate the funds (e.g., a send money transaction with an unregistered customer), the *account\_id* parameter should be passed in. For more information, see [Table 4-1, “Send money API prepare request parameters,” on page 7](#).

---

**Note:** To access wallet account IDs, navigate to **My Account > Settings > Developer Settings**.

---

## 2.3. Contact for queries

For all merchant support, please contact the Skrill Merchant Service Department: Email: [merchantservices@skrill.com](mailto:merchantservices@skrill.com)

**Table 2-1: Contact Numbers**

Language	Telephone Number	Operating Times (weekdays)
English	+44 (0) 2083387760	8am - 5pm GMT
English US	+1 855 6225 167	8am - 6pm EST

---

## 3. SECURITY

---

This section describes the following:

- *Security requirements*
- *Separate API/MQI password*
- *Separate API/MQI password*
- *Secret word*

### 3.1. Security requirements

All requests to the Automated Payments Interface (API) and Merchant Query Interface (MQI) must be standard HTTPs GET or POST requests; all endpoints accept both methods. The HTTPs protocol provides a secure means of verifying the program on the client host. Plain text HTTP requests are forbidden, and if the client sends an HTTP request to the server it will be denied.

- Skrill recommends using POST for maximum security.
- Do not mix GET and POST requests. Choose which method to use and apply consistently.
- Do not mix GET and POST calls. Choose a preferred method and use that for all MQI/API calls.
- POST parameters are encoded using Content-Type: application/x-www-form-urlencoded.
- GET parameters are sent as part of the URL query string, for example: `https://www.skrill.com/app/query.pl?action=status_trn&email=mb654@abv.bg&password=53903d217504eb37f3fdb0ce77610558&mb_trn_id=104627261`.

---

**Note:** If you currently do not send HTTPs headers for tracking reasons, you should be aware that this can be used as a loophole by potential website hackers.

---

#### 3.1.1. Accept request-header field

The *Accept request-header field* can be used to specify certain media types that are acceptable for the response to your HTTP request. If you are using Accept request headers in your API requests, please implement as follows:

- Use \*/\* or text/\* or text/html Accept Headers for MQI functions
- Use \*/\* or text/\* or text/xml Accept Headers for Automated Payment Interface functions



### 3.1.2. Content types

The following content types are returned in the response for all API calls:

```
text/xml
```

The following content types are returned in the response for all MQI calls (excluding account history):

```
text/html
```

Account history returns the following content types:

```
application/vnd.ms-excel; charset=UTF-8
```

## 3.2. Separate API/MQI password

You will need to enable the MQI (merchant query interface) and API (automated payment interface) and set up an MQI/API password to use 1-Tap. You can also change the MQI/API password here in this section.

To enable the MQI and/or API:

1. Log in to your Skrill merchant account at [www.skrill.com](http://www.skrill.com)
2. Go to **Settings > Developer Settings** ([Figure 3-1](#))
3. Set API / MQI Password using the toggle, type and confirm by clicking **Save**
4. For each section, specify the IP address(es) or IP address range of your Shopify server. This prevents payment or money transfer requests from other IP addresses if your secret word or API password is compromised. All requests from other IP addresses are denied. Access can be granted to:
  - A single IP address (e.g. 145.76.160.206 using the example above)
  - Multiple IP addresses, separated by space (e.g. 192.168.0.2 10.0.0.2)
  - A subnet in CIDR notation (e.g. 192.168.0.0/24)
5. Activate the API/MQI using the toggle
6. Set Secret Word using the toggle, type and confirm by clicking the **Save** button
 

**Note:** Your Secret Word must contain at least: 8 characters, 1 uppercase letter, 1 lowercase letter, and 1 number.

---

**Warning:** Set Secret Word using the toggle, type and confirm by clicking the **Save** button

**Note:** Your Secret Word must contain at least: 8 characters, 1 uppercase letter, 1 lowercase letter, and 1 number. CIDR ranges should be no longer than 256 IP addresses.

---



---

**Note:** If the **Settings > Developer Settings** section is not displayed in your account, contact [Skrill Merchant Services](#)

---

**Figure 3-1: Enable the API and MQI and set up a password and IP range for these services**

**Skrill** API GUIDES [Help](#)

Home > Settings

**Developer Settings**  
Configure API and security settings for your wallet account

Bla Bla Wallet  
ID: 116404435

Developer Settings : Bla Bla Wallet

Wallet IP Whitelisting **API / MQI / GSR / CVT Management** API Documentation

☐ **Automated Payment Interface (API)** Disabled

Allow requests to the API only from the following IP addresses.

Your API secure IP addresses:

87.120.141.129

☐ **Merchant Query Interface (MQI)** Disabled

Allow requests to the MQI only from the following IP addresses.

Your MQI secure IP addresses:

No active secure IP address found

☐ **Gateway Secure Restriction (GSR)** Disabled

Allow gateway sessions to be generated only from the following IP addresses.

Your GSR secure IP addresses:

5.5.5.5

☐ **Set API / MQI Password** Enabled

Please provide API/MQI Password in order to enable the service.

Current Password

New Password

Confirm Password

Cancel Save

☐ **Set Secret Word** Enabled

Please provide Secret Word in order to enable the service.

New Password

Confirm Password

Cancel Save

[Signs](#) | [Privacy Notice](#) | [Regulatory Disclosures](#) | [Cookie Notice](#)

© 2020 Paysafe Holdings UK Limited. All rights reserved. Skrill is a registered trademark of Paysafe Holdings UK Limited. Skrill Limited is registered in England and Wales with company number 9038907 and its registered office at 25 Canada Square, London E14 9QJ. Authorized and regulated by the Financial Conduct

## 4. SEND MONEY USING AN HTTPS REQUEST

You can make mass payments using the Skrill Automated Payments Interface (API). This offers the same functionality that is available on My Account, but it allows you to automate the sending of payment details from your servers to Skrill using an HTTPS request.

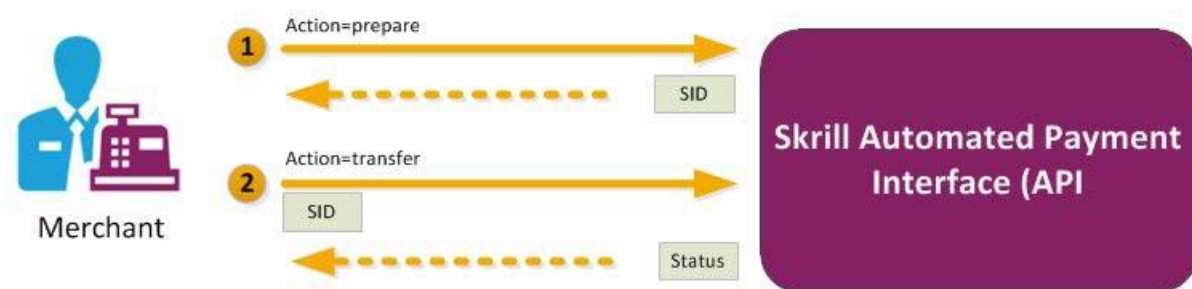
Automated payment transfers are implemented by sending an HTTPS request to the following URL:

<https://www.skrill.com/app/pay.pl>.

The process consists of two steps:

- Sending a transfer prepare request to initiate a *session*, which lasts 15 minutes
- Executing the transfer within the session

After each step Skrill returns an XML response that contains the result of the performed action. See the example below.



**Figure 4-1: Steps in the Transfer request**

Using a multi-currency account, Skrill automatically debits the account matching the currency of the send money request. If the currency specified in the request is not supported by your merchant account, the primary currency account will be debited.

### 4.1. Sending a transfer prepare request

Query parameter: *action=prepare*

You must include the parameters described below in your HTTPS request.

**Table 4-1: Send money API prepare request parameters**

Parameter	Description	Required?	Example value
action	The required action. In the first step, this is ' <i>prepare</i> '.	Yes	action=prepare
email	Your merchant account email address.	Yes	info@merchant.com
password	Your MD5 API/MQI password.	Yes	9f535b6ae672f627e4a5f79f2b7c63fe
amount	Amount to be transferred.	Yes	10.95

**Table 4-1: Send money API prepare request parameters (Continued)**

Parameter	Description	Required?	Example value
currency	Currency. For a list of currencies accepted by Skrill, see <a href="#">Table 8-1 on page 32</a> .	Yes	EUR
bnf_email	Recipient's (beneficiary's) email address.	Yes	customer@host.com
subject	Subject of the notification email. Up to 250 1-byte characters.	Yes	Your order is ready
note	Comment to be included in the notification email. Up to 2000 1-byte characters.	Yes	Details are available on our website.
frn_trn_id	Your reference ID (must be unique if submitted).	No	A1234
account_id	Your currency account ID. For more information, see <a href="#">Separating balances by fund origin, on page 3</a>	Yes/No*	1234567

\* The **account\_id** parameter is required when Skrill is unable to automatically reserve funds from the appropriate currency account (EEA or non-EEA), e.g., when you are sending money to an unregistered/US customer. The **account\_id** parameter specifies which currency account should be selected.

### Skrill response

Skrill returns an XML response to your request, which contains a **<response>** element containing one of the following elements:

- **<sid>** element - returned if the authorisation and payment preparation is successful. The SID (Session Identifier) must be submitted in your transfer execution request (see [Table 4-2 on page 9](#)).
- **<error>** element – included if an error occurs. It includes an **<error\_msg>** element, which contains the error message description.

### Example 1: Successful prepare request

Request:

```
GET https://www.skrill.com/app/
pay.pl?action=prepare&email=merchant@host.com&password=6b4c1ba48880bcd3341dbaeb68b2
647f&amount=1.2&currency=EUR&bnf_email=beneficiary@domain.com&subject=some_subject&
note=some_note&frn_trn_id=111
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <sid>5e281d1376d92ba789ca7f0583e045d4</sid>
</response>
```

## Example 2: Failed prepare request

This example shows a request that failed, due to a missing *'amount'*.

Request:

```
GET https://www.skrill.com/app/
pay.pl?action=prepare&email=merchant@host.com&password=6b4c1ba48880bcd3341dbaeb68b2
647f&currency=EUR&bnf_email=beneficiary@domain.com&subject=some_subject&note=some_n
ote&frn_trn_id=111
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <error>
    <error_msg>MISSING_AMOUNT</error_msg>
  </error>
</response>
```

## Example 3: Failed with error LOCK\_LEVEL\_9

This example shows a request that failed because a Skrill Wallet merchant account with 1-Tap enabled is locked following more than three failed login attempts.

Request:

```
GET https://www.skrill.com/app/
pay.pl?action=prepare&email=merchant@host.com&password=6b4c1ba48880bcd3341dbaeb68b2
647f&amount=500&currency=INR&bnf_email=beneficiary@domain.com&subject=some_subject&
note=some_note&frn_trn_id=111
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <error>
    <error_msg>LOCK_LEVEL_9</error_msg>
  </error>
</response>
```

To clear the error, reset the password for the merchant account; and then reconfigure your API/MQI password, which is the MD5 hash of your merchant password, before making any further prepare requests. In some situations you will not be able to reset the password, in which case you should contact [merchantservices@skrill.com](mailto:merchantservices@skrill.com).

## 4.2. Executing a transfer request

Query parameter: *action=transfer*

Your web servers should include the SID information provided in the XML response from Skrill in the transfer execution request, as described below.

**Table 4-2: Execute transfer request parameters**

Field	Description	Required?	Example value
action	The required action. In the second step, this is <i>'transfer'</i> .	Yes	action=transfer
sid	Session identifier returned in response to the prepare request.	Yes	5e281d1376d92ba789ca7f0583e045d4

## Skrill response

The correct XML response contains a **<response>** element that includes the following elements:

- **<transaction>** element – returned if the transfer is successful; the response includes the elements described in **Table 4-3** below.
- **<error>** element – returned if an error occurs, which includes an **<error\_msg>** element, which provides details of the error.

**Table 4-3: Successful response to transfer request**

Field	Description
Amount	Amount paid in the currency of your Skrill account.
Currency	Currency of your Skrill account.
Id	Transaction ID.
Status	Numeric value of the transaction status: 1 – scheduled (if beneficiary is not yet registered at Skrill) 2 - processed (if beneficiary is registered)
status_msg	Text value of the transaction status.

## Example of a successful transfer request

Request:

```
GET https://www.skrill.com/app/
pay.pl?action=transfer&sid=5e281d1376d92ba789ca7f0583e045d4
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <transaction>
    <amount>1.20</amount>
    <currency>EUR</currency>
    <id>497029</id>
    <status>2</status>
    <status_msg>processed</status_msg>
  </transaction>
</response>
```

### 4.3. Reposting a transfer request

If there is a communication error during the transfer, you must resend the transfer request within 15 minutes of the previous request, before the session expires.

---

**Note:** The Skrill server executes only one transaction per session, so the request cannot be duplicated.

---

The outcome of reposting a transfer request depends on the transfer execution status:

- If the transaction has already been executed within this session you will need to generate a new session ID (action=prepare), since only one transaction is allowed per session.
- If the transaction associated with this session is still being executed, Skrill responds with status ***EXECUTION\_PENDING***. In this case you do not need to generate a new session ID and can wait for the response.

## 5. QUICK CHECKOUT PAYOUTS

Quick Checkout payouts enable Skrill gambling merchants to transfer funds (such as winnings) to their customers by referencing an original QCO transaction (payment) made on the gateway. Since the money is sent directly to the payment instrument used for the original payment, the customer does not need a Skrill Wallet account.

### Restrictions:

Quick Checkout payouts functionality is available only for:

- Rapid Transfer payments
- NETELLER payments

Merchants wishing to use the Quick Checkout payouts facility need to have this functionality enabled on their account by the [Merchant Services](#) team.

Quick Checkout payouts use an extension to the Automated Payments Interface (API), and are implemented by sending an HTTPs request to the following URL:

<https://www.skrill.com/app/pay.pl>

The process consists of two steps:

- Sending a payout prepare request to initiate a *session*
- Executing the Quick Checkout payout within the session

After each step Skrill returns an XML response that contains the result of the performed action.

### 5.1. Sending a payout prepare request

Query parameter: *action=prepare*

You must include the parameters described below in your HTTPs request.

**Table 5-1: Payout prepare request parameters**

Parameter	Description	Required?	Example value
action	The required action. In the first step, this is <i>prepare</i> .	Yes	action=prepare
email	Your email address.	Yes	info@merchant.com
password	Lower-case version of your MD5 API/MQI password.	Yes	9f535b6ae672f627e4a5f79f2b7c63fe
transaction_id	The transaction ID of the original payment.	Yes/No *	A500123
mb_transaction_id	The Skrill transaction ID of the original payment.	Yes/No *	4585262
amount	Amount to be transferred.	Yes	10.95



**Table 5-1: Payout prepare request parameters (Continued)**

Parameter	Description	Required?	Example value
currency	Currency. For a list of currencies accepted by Skrill Quick Checkout Payouts, see <b>Table 8-1 on page 32</b> .	Yes	EUR
subject	Subject of the notification email. Up to 250 1-byte characters.	Yes	Your order is ready
note	Comment to be included in the notification email. Up to 2000 1-byte characters.	Yes	Details are available on our website.
frn_trn_id	Your reference ID (must be unique if submitted).	No	A1234
merchant_fields	A comma-separated list of up to five field names that are passed back to your server when the payment is confirmed.	No	Field1, Field2
Field1	An additional field that you can include containing your own unique parameters	No	Value1
Field2	An additional field that you can include containing your own unique parameters	No	Value2
status_url	<p>URL or email address to which status updates should be sent. The following ports can be used for this URL:</p> <p>80, 81, 82, 83, 88, 90, 178, 419, 433, 443, 444, 448, 451, 666, 800, 888, 1025, 1430, 1680, 1888, 1916, 1985, 2006, 2221, 3000, 4111, 4121, 4423, 4440, 4441, 4442, 4443, 4450, 4451, 4455, 4567, 5443, 5507, 5653, 5654, 5656, 5678, 6500, 7000, 7001, 7022, 7102, 7777, 7878, 8000, 8001, 8002, 8011, 8014, 8015, 8016, 8027, 8070, 8080, 8081, 8082, 8085, 8086, 8088, 8090, 8097, 8180, 8181, 8443, 8449, 8680, 8843, 8888, 8989, 9006, 9088, 9443, 9797, 10088, 10443, 12312, 18049, 18079, 18080, 18090, 18443, 20202, 20600, 20601, 20603, 20607, 20611, 21301, 22240, 26004, 27040, 28080, 30080, 37208, 37906, 40002, 40005, 40080, 50001, 60080, 60443</p>	No	<a href="https://www.merchant.com/update.cqi">https://www.merchant.com/update.cqi</a>

**Notes:**

- \* You must submit either *transaction\_id* or *mb\_transaction\_id*.
- If you accidentally include a *bnf\_email* parameter in your prepare request (which is used when preparing to send money, as described on [page 8](#)), the subsequent transfer action sends money to a Skrill balance account instead of making a Quick Checkout payout.

**Skrill response**

Skrill returns an XML response to your request, which contains a *<response>* element containing one of the following elements:

- <sid>* element - returned if the authorisation and payment preparation is successful. The SID (Session Identifier) must be submitted in your transfer execution request (see [Table 5-2 on page 14](#)).
- <error>* element – included if an error occurs. It includes an *<error\_msg>* element, which contains the error message description. For an explanation of this message, see [Table 8-2](#) and [Table 8-3](#).

**Example prepare payout request**

```
GET https://www.skrill.com/app/
pay.pl?action=prepare&email=merchant@host.com&password=8bcf2ad23abdc7f4b8e65efaadf7
1e19&amount=10&currency=AED&subject=subject&note=note&frn_trn_id=my-frn-trn-
1&mb_transaction_id=194573960
```

**Example success response**

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <sid>5e281d1376d92ba789ca7f0583e045d4</sid>
</response>
```

**Example error response**

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <error>
    <error_msg>MISSING_AMOUNT</error_msg>
  </error>
</response>
```

## 5.2. Executing a payout transfer request

Query parameter: *action=transfer*

Your web servers should include the SID information provided in the XML response from Skrill in the payout transfer execution request, as described below.

**Table 5-2: Payout transfer request parameters**

Field	Description	Required?	Example value
action	The required action. In the second step, this is the payout ' <i>transfer</i> '.	Yes	action=transfer
sid	Session identifier returned in response to the prepare request.	Yes	5e281d1376d92ba789ca7f0583e045d4

## Skrill response

The XML response contains a **<response>** element that includes either of the following elements:

- **<transaction>** element – returned if the payout transfer is successful; the response includes the elements described in **Table 5-3** below.
- **<error>** element – returned if an error occurs, and includes an **<error\_msg>** element, which provides details of the error. This is shown in **Table 5-3**.

**Table 5-3: Response to transfer request**

Element	Description
amount	Amount paid in the currency of your Skrill account.
currency	Currency of your Skrill account.
id	Skrill transaction ID.
status	Numeric value of the transaction status: -2 = failed 2 = processed 0 = pending
status_msg	Textual transaction status.
error_msg	<ul style="list-style-type: none"> <li>• BALANCE_NOT_ENOUGH - payout amount exceeds account balance</li> <li>• ORIGINAL_PAYMENT_NOT_PROCESSED - referenced transaction was not processed</li> <li>• ORIGINAL_PAYMENT_UNSUPPORTED_FOR_PAYOUT - payment method used for the referenced transaction is not supported for payouts</li> <li>• GENERIC_ERROR - other errors not covered above</li> </ul>

## Example successful payout request

```
GET https://www.skrill.com/app/
pay.pl?action=transfer&sid=5e281d1376d92ba789ca7f0583e045d4
```

## Example successful response

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <transaction>
    <amount>1.20</amount>
    <currency>EUR</currency>
    <id>497029</id>
    <status>2</status>
    <status_msg>processed</status_msg>
  </transaction>
</response>
```

### 5.3. Payout status report

If the payment transfer API returns a status of *pending* (status = 0), then whenever the status of the transaction changes, Skrill sends details of the change to the *status\_url* provided in your payment request using a standard HTTP POST. The Skrill server continues to post the status until it receives an HTTP OK (200) response from your server, or until it has posted more than 10 times. The table below shows the parameters sent to your *status\_url* page:

**Table 5-4: Payout Status Report**

Field	Description	Required	Example
transaction_id	Your reference ID (the <i>frn_trn_id</i> from the request)	Yes/No	500123
mb_transaction_id	Skrill's internal unique reference ID for this transaction	Yes	5585262
merchant_id	Unique ID of your Skrill account; needed to calculate the MD5 signature.	Yes	1392345
amount	Payment amount posted in your HTTP request	Yes	2
currency	Payment currency posted in your HTTP request	Yes	EUR
status	2 = Processed -2 = Failed	Yes	2
mb_amount	Amount in your Skrill account currency	Yes	9.99
mb_currency	Skrill account currency	Yes	EUR
md5sig	Upper-case MD5 signature. See <i>MD5 signature, on page 17</i> .	Yes	9F535B6AE672F527E 4A5F79F2B7C63FE
sha2sig	Upper-case Sha2 signature. This is constructed in the same way as the MD5 signature, but with a different hashing algorithm. This parameter is not available by default. To enable this option, send a request to <a href="mailto:merchantservices@skrill.com">merchantservices@skrill.com</a> .	No	DBB7101322257A31 1F0SD1C527053058F C7E464E30 BCFB4613F09053C22 DD1F8
pay_to_email	Your email address	Yes	info@merchant.com
pay_from_email	Email address of the customer to whom the funds are being sent	Yes	payer123@skrill.com

### 5.3.1. MD5 signature

The **md5sig** parameter consists of an MD5 sum on a string built by concatenating the following parameters and converting the result to upper-case:

**Table 5-5: MD5 signature parameters**

Value	Description	Example
merchant_id	Your Skrill account user ID	4637827
mb_transaction_id	The new Skrill transaction ID	5585262
MD5 of secret word	The upper-case MD5 value of the secret word submitted in the <b>Settings &gt; Developer Settings</b> section of your Skrill account.	327638C253A4637199CEBA6642371F20
mb_amount	Amount refunded in the currency of your account.	9.99
mb_currency	Currency of your account.	EUR
status	The status of the refund transaction.	2

#### Example code

Concatenated fields in Ruby code:

```
fields = [merchant_id, mb_transaction_id, Digest::MD5.hexdigest(secret).upcase,
mb_amount, mb_currency, status].join
md5sig == Digest::MD5.hexdigest(fields).upcase
```

Using the example values in the table **Table 5-5**, the following MD5 code is returned:

```
CF9DCA614656D19772ECAB978A56866D
```

#### Example status report

```
merchant_id=290186320&transaction_id=200366670&mb_transaction_id=200366670&mb_amount=74.218786&mb_currency=GBP&status=2&md5sig=3ED76725C3E3CE6CE25F16F01BDFDF1D&amount=80.0&pay_from_email=payer%40gmail.com&pay_to_email=merchant%40info.com&currency=EUR
```

## 6. REFUNDS

**Note:** Refunds are not available for gambling and Forex merchants

You can use the Automated Payments Interface to make automated partial or full refunds to customers, up to the amount of the original payment.

You must send your HTTPS refund request to the following URL:

<https://www.skrill.com/app/refund.pl>

The refund is made in two steps:

- Preparation of the refund
- Execution of the refund

### 6.1. Preparing a refund

Query parameter: *action=prepare*

The following parameters must be included in the refund prepare request:

**Table 6-1: Refund preparation parameters**

Field	Description	Required	Example
action	Defines the prepare step of the refund request.	Yes	action=prepare
email	Your email address.	Yes	info@merchant.com
password	The MD5 of your API/MQI password. Note: only the lower-case of the MD5 value is accepted.	Yes	9f535b6ae672f627e4a5f79f2b7c63fe
transaction_id	Your transaction ID to be refunded.	Yes/No *	500123
mb_transaction_id	The Skrill transaction ID to be refunded.	Yes/No *	4585262
amount	Amount to refund in the currency used by the merchant account. This field is only used for partial refunds.	No	9.99
refund_note	Refund note sent to the customer. This note forms part of the email sent to the customer to inform them that they have received a refund.	No	Product no longer in stock
merchant_fields	A comma-separated list of field names that are passed back to your server when the refund payment is confirmed (maximum 5 fields).	No	Field1,Field2

**Table 6-1: Refund preparation parameters (Continued)**

Field	Description	Required	Example
Field1	An additional field you can include, containing your own unique parameters.	No	Value1
Field2	An additional field you can include, containing your own unique parameters.	No	Value2
refund_status_url	<p>URL or email address to which status updates should be sent.</p> <p>The following ports can be used for this URL:</p> <p>80, 81, 82, 83, 88, 90, 178, 419, 433, 443, 444, 448, 451, 666, 800, 888, 1025, 1430, 1680, 1888, 1916, 1985, 2006, 2221, 3000, 4111, 4121, 4423, 4440, 4441, 4442, 4443, 4450, 4451, 4455, 4567, 5443, 5507, 5653, 5654, 5656, 5678, 6500, 7000, 7001, 7022, 7102, 7777, 7878, 8000, 8001, 8002, 8011, 8014, 8015, 8016, 8027, 8070, 8080, 8081, 8082, 8085, 8086, 8088, 8090, 8097, 8180, 8181, 8443, 8449, 8680, 8843, 8888, 8989, 9006, 9088, 9443, 9797, 10088, 10443, 12312, 18049, 18079, 18080, 18090, 18443, 20202, 20600, 20601, 20603, 20607, 20611, 21301, 22240, 26004, 27040, 28080, 30080, 37208, 37906, 40002, 40005, 40080, 50001, 60080, 60443</p>	No	https:// www.merchant.com/ refund_update.cqi

**Notes:**

- \* You must submit either '*transaction\_id*' or '*mb\_transaction\_id*'.
- If no '*amount*' value is submitted, the refund will be for the full amount of the original transaction.
- If the '*refund\_note*' value is submitted, it is shown in the body of the notification email sent to the customer.

**XML server response**

The resulting XML contains a **<response>** element with one of the following elements, depending on the success of the operation:

- **<sid>** (session identifier) element – returned if the prepare request is successful. This must be submitted with the **refund** action in the next step:

```
<response>
  <sid>4504848cb1ed0d29f60458bf992399fd</sid>
</response>
```

- **<error>** element – returned if an error occurs. Contains an **<error\_msg>** element containing the error message:

```
<response>
  <error>
    <error_msg>CANNOT_LOGIN</error_msg>
  </error>
</response>
```

## Error messages

See the table below for details of error messages.

**Table 6-2: Refund preparation error messages**

Error message	Description
CANNOT_LOGIN	Invalid combination of email and password is supplied.
INVALID_EMAIL	An Invalid 'email' parameter is supplied.
INVALID_OR_MISSING_ACTION	The 'action' parameter is not supplied in the query.
LOGIN_INVALID	Missing 'email' or 'password' parameters.
NO_LOGIN_EXPLANATION	Merchant using non-existent email address.
REFUND_DENIED	Refund feature is not activated.

## Prepare refund examples

Successful prepare request:

```
POST https://www.skrill.com/app/refund.pl
Content-Type: application/x-www-form-urlencoded
action=prepare&email=info@merchant.com&password=9f535b6ae672f627e4a5f79f2b7c63fe&transaction_id=500123&amount=9.99&refund_note=example_note&refund_status_url=https://www.merchant.com/refund_update.cgi&merchant_fields=Field1,Field2&Field1=Value1&Field2=Value2
```

Successful prepare response:

```
<response>
  <sid>d831e9072e8b89c57a3654ddf5fcb907</sid>
</response>
```

Incorrect request (invalid merchant API/MQI password):

```
POST https://www.skrill.com/app/refund.pl?action=prepare&email=info@merchant.com&password=9f535b6ae672f627e4a5f79f2b7c64fe&amount=9.99&refund_note=example_note&refund_status_url=https://www.merchant.com/refund_update.cgi&merchant_fields=Field1,Field2&Field1=Value1&Field2=Value2
```

Response:

```
<response>
  <error>
    <error_msg>CANNOT_LOGIN</error_msg>
  </error>
</response>
```



## 6.2. Executing a refund

Query parameter: *action=refund*

The following parameters must be included in the refund execution request:

**Table 6-3: Refund execution parameters**

Field	Description	Required	Example
action	Defines the execution step of the refund request.	Yes	action=refund
sid	Session identifier returned in response to the prepare request.	Yes	d831e9072e8b89c57a3654d df5fcb907

### XML Server Response

The server returns XML containing a *<response>* element, which includes some of the following elements, depending on the success of the operation:

**Table 6-4: Refund response**

Element	Description		Example
mb_amount	Amount refunded in the currency of your Skrill account.		9.99
mb_currency	Currency of your Skrill account.		EUR
transaction_id	Your refund transaction ID as submitted in the request.		500123
mb_transaction_id	The Skrill transaction ID for the refund.		5585262
Field1	The first additional field pre-defined in ' <i>merchant_fields</i> ' parameter		Value1
Field2	The second additional field pre-defined in ' <i>merchant_fields</i> ' parameter		Value2
status	2 = processed, 0 = pending, -2 = failed.		2
error	BALANCE_NOT_ENOUGH	Refund amount exceeds account balance.	CC_REFUND_FAILED
	CC_REFUND_FAILED	Refund to a credit/debit card failed.	
	RESERVE_EXCEEDED	Refund amount is blocked by rolling/fixed reserve.	
	GENERIC_ERROR	Other errors, different from those described above. Note this error code is also used if this transaction has already been refunded.	

## Refund execution example

### Request:

```
GET https://www.skrill.com/app/
refund.pl?action=refund&sid=d831e9072e8b89c57a3654ddf5fcb907
```

### Response:

```
<response>
  <mb_amount>2</mb_amount>
  <mb_currency>EUR</mb_currency>
  <mb_transaction_id>381526883</mb_transaction_id>
  <Field1>Value1</Field1>
  <Field2>Value2</Field2>
  <status>2</status>
  <transaction_id/>
</response>
```

## 6.3. Refund status report

If a request for refund cannot be executed at the moment, Skrill sends a response with status '0' (pending) in the prepare step. When Skrill receives an update on the status of the refund, a notification is sent to your *refund\_status\_url* page.

If you would like to receive notifications for every status of your refund request, contact the [Merchant Services](#) team.

This status report consists of the following fields:

**Table 6-5: Status report fields**

Field	Description	Required	Example
transaction_id	Your transaction ID for the refund as submitted in the request.	Yes /No	500123
mb_transaction_id	The Skrill transaction ID for the refund.	Yes	5585262
status	2 = processed, -2 = failed.	Yes	2
mb_amount	Amount refunded in the currency of your Skrill account	Yes	9.99
mb_currency	Currency of your Skrill account.	Yes	EUR
md5sig	Upper-case MD5 signature. See <i>MD5 signature</i> below.	Yes	9F535B6AE672F627E4A5F79F2B7C63FE
sha2sig	Upper-case Sha2 signature. This is constructed in the same way as the MD5 signature, but with a different hashing algorithm.  This parameter is not available by default. To enable this option, send a request to <a href="mailto:merchantservices@skrill.com">merchantservices@skrill.com</a>	No	DBB7101322257A311F08D1C527053058FC7E464E30BCFB4613F09053C22DD1F8

## MD5 signature

The '*md5sig*' parameter consists of an MD5 sum on a string built by concatenating the following parameters and converting the result to upper-case

**Table 6-6: MD5 Signature parameters**

Value	Description	Example
merchant_id	Your Skrill account user ID.	4637827
mb_transaction_id	The new Skrill transaction ID for refund.	5585262
MD5 of secret word	The upper-case MD5 value of the secret word submitted in the <b>Settings &gt; Developer Settings</b> section of your Skrill account.	327638C253A4637199C EBA6642371F20
mb_amount	Amount refunded in the currency of your account.	9.99
mb_currency	Currency of your account.	EUR
status	The status of the refund transaction.	2

## Example code

Concatenated fields in Ruby code:

```
fields = [merchant_id, mb_transaction_id, Digest::MD5.hexdigest(secret).upcase,
mb_amount, mb_currency, status].join
md5sig == Digest::MD5.hexdigest(fields).upcase
```

Using the example values in **Table 6-6** above, the following MD5 code is returned:

```
CF9DCA614656D19772ECAB978A56866D
```

---

## 7. MERCHANT QUERY INTERFACE

---

The Merchant Query Interface allows you to query the Skrill database for the current status of your transactions as well as perform actions connected to Skrill 1-Tap and recurring payments. You can access the MQI by posting an HTTPS query to:

<https://www.skrill.com/app/query.pl>

Each MQI query requires the three general parameters *email*, *password* and *action*, and other parameters specific to the requested action (see *MQI Actions* below).

**Table 7-1: General query parameters**

Field Name	Description	Required?	Example value
email	The email address of your Skrill account.	Yes	info@merchant.com
password	The lower-case hex MD5 of your API/MQI password.	Yes	9f535b6ae672f627e4e5f79f2b7c63fe
action	The required action.	Yes	repost

### 7.1. MQI Actions

The following MQI actions are supported:

- *Repost transaction status*
- *View transaction status*
- *View account history*
- *Cancel a recurring payment*
- *View recurring payment status*
- *Extend the end date of a recurring payment*
- *Cancel a Skrill 1-Tap payment*
- *View Skrill 1-Tap payment status*
- *404\t\tillegal parameter value: 17651712043534553\n*

### 7.1.1. Repost transaction status

Query parameter: ***action=repost***

This action allows you to request a repost of the status of a transaction to your ***status\_url*** page.

---

**Note:** This is a repost of the same status report that was posted when the payment was made, and is sent to the same status URL that was specified in the original payment.

---

In response, Skrill posts a status report (for details, refer to the [Skrill Quick Checkout Integration Guide](#) or [Skrill Wallet Checkout Integration guide](#) as appropriate). If no status report was posted initially, this action will return a ***'403 Transaction not found: TRN\_ID'*** error.

The parameters listed below are required.

**Table 7-2: Repost parameters**

Field Name	Description	Required?	Example value
trn_id	Your transaction ID	Yes/No	500123
mb_trn_id	Skrill transaction ID	Yes/No	4585262
status_url	Where to post the notification	No	https://www.merchant.com/mb_notifications.asp

**Notes:**

- Either ***trn\_id*** or ***mb\_trn\_id*** must be supplied. If both are given, ***trn\_id*** will be used.
- If ***status\_url*** is not provided, the ***status\_url*** given at the time the transaction was created will be used.
- For a successful HTTP request, the HTTP response body ***200\t\tOK\n\n*** is returned (using escape sequences to represent special characters).

### 7.1.2. View transaction status

Query parameter: ***action=status\_trn***

This action gives a direct response with the status of the payment. It includes the same details as in the ***'repost'*** action, but sends a direct response to the request rather than to the old status URL. The following parameters are required:

**Table 7-3: Transaction status parameters**

Field Name	Description	Required?	Example value
trn_id	Your transaction ID.	Yes/No	500123
mb_trn_id	Skrill transaction ID.	Yes/No	4585262

**Notes:**

- Either ***trn\_id*** or ***mb\_trn\_id*** must be supplied and if both are given, ***trn\_id*** will be used.

- If a transaction with the given ID is found, the response will be a query string that contains the transaction details. The string is encoded using the '*application/x-www-form-urlencoded*' format.

### 7.1.2.1.Examples

#### API transaction

##### Request:

```
GET https://www.skrill.com/app/
query.pl?action=status_trn&email=mb654@abv.bg&password=53903d217504eb37f3fdb0ce7761
0558&mb_trn_id=104627261
```

##### Response:

```
200 -> -> OK
status=2&merchant_id=6999381&mb_transaction_id=104627261&mb_amount=1.2&pay_to_email
=mb654%40abv.bg&currency=BGN&amount=2.346996&transaction_id=&pay_from_email=test%40
test.bg&mb_currency=EUR
```

The symbol -> is used to indicate a tab character.

Using escape sequences to represent special characters, the response is as follows:

```
200\t\tOK\n
status=2&merchant_id=6999381&mb_transaction_id=104627261&mb_amount=1.2&pay_to_email
=mb654%40abv.bg&currency=BGN&amount=2.346996&transaction_id=&pay_from_email=test%40
test.bg&mb_currency=EUR
```

#### Payment Gateway transaction

##### Request:

```
GET https://www.skrill.com/app/
query.pl?action=status_trn&email=merchant@host.com&password=53903d217504eb37f3fdb0c
e77610558&mb_trn_id=104441110
```

##### Response:

```
200 -> -> OK
status=2&Field1=TR234567&md5sig=6AB68D3465F57492B7412ED0EB013621&merchant_id=999998
1&pay_to_email=merchant%40host.com&mb_amount=33.24911&mb_transaction_id=101149910&c
urrency=EUR&amount=17&transaction_id=49989810fa3ed45c&pay_from_email=payeremail%40h
ost.bg&mb_currency=BGN
```

The symbol -> is used to indicate a tab character.

Using escape sequences to represent special characters, the response is as follows:

```
200\t\tOK\n
status=2&Field1=TR234567&md5sig=6AB68D3465F57492B7412ED0EB013621&merchant_id=999998
1&pay_to_email=merchant%40host.com&mb_amount=33.24911&mb_transaction_id=101149910&c
urrency=EUR&amount=17&transaction_id=49989810fa3ed45c&pay_from_email=payeremail%40h
ost.bg&mb_currency=BGN
```

### 7.1.3. View account history

Query parameter: ***action=history***

You can use the '***history***' action to request a list of all your transactions for a specified period. The following parameters are required:

**Table 7-4: History parameters**

Field Name	Description	Required?	Example value
start_date	The start date in <b><i>DD-MM-YYYY</i></b> format.	Yes	29-05-2016
end_date	The end date in <b><i>DD-MM-YYYY</i></b> format.	No	30-06-2017
account_id	Your currency account ID. For more information, see <b><i>Separating balances by fund origin, on page 3.</i></b>	Yes/No	1234567

#### Notes:

- Upon success, Skrill returns the complete account history for the specified period in CSV (comma separated values) format.
- If the ***end\_date*** parameter is not specified, Skrill uses today's date.

#### Extended transaction history feature

It is possible to request the extended currency account information for each transaction in the report. This information is presented in three additional columns: ***currency***, ***region***, and ***account\_id***.

By default, these features are not enabled for merchant accounts. If you would like to have them enabled for your account, please contact the Skrill Merchant Service Department at: [merchantservices@skrill.com](mailto:merchantservices@skrill.com)

---

**Note:** If you are using a multi-currency merchant account, we recommend that you enable the extended transaction history feature, otherwise, only the transaction history for the primary currency will be returned.

---

#### 7.1.3.1.Example

Request:

```
GET https://www.skrill.com/app/
query.pl?email=merchant@host.com&password=53903d217504eb37f3fdb0ce77610558&action=h
istory&start_date=25-05-2017&end_date=25-06-2017
```

Response:

```
csv file: mb_history.csv
```

### 7.1.4. Cancel a recurring payment

Query parameter: ***action= cancel\_rec***

This action allows you to cancel a recurring payment. The following parameters are required:

**Table 7-5: Cancel parameters**

Field Name	Description	Required?	Example value
trn_id	Your transaction ID.	Yes	500123

For a successful cancel request, the HTTP response body ***200\t\tOK\n\n*** is returned (using escape sequences to represent special characters)

### 7.1.5. View recurring payment status

Query parameter: ***action=status\_rec***

This action allows you to check the status of a recurring payment. The following parameters are required:

**Table 7-6: Recurring payment status parameters**

Field Name	Description	Required?	Example value
trn_id	Your transaction ID.	Yes	500123

If a transaction with the given ID is found, the response contains the following parameters:

- Status: ***0*** = active, ***-1*** = cancelled, ***-2*** = failed, ***1*** = finished
- Next payment date in ***dd-mm-yyyy*** format. This parameter is returned only if status is '***active***' or '***failed***'
- End date in ***dd-mm-yyyy*** format. This parameter is returned only if status is '***active***' or '***failed***'

#### 7.1.5.1. Recurring payment status example

Request:

```
GET https://www.skrill.com/app/
query.pl?action=status_rec&email=merchant@host.com&password=2813F1526CD435D296A2A8F
EE37889AD&trn_id=yourtansID123
```

Response:

```
200 -> -> OK
Status: 0 Next payment date: 26-05-2017, End date: 26-12-2017
```

The ***->*** symbol represents a tab character.

Using escape sequences to represent special characters, the response is as follows:

```
200\t\tOK\nStatus: 0 Next payment date: 26-05-2017, End date: 26-12-2017
```



### 7.1.6. Extend the end date of a recurring payment

Query parameter: *action= extend\_rec*

This action allows you to extend the end date (*rec\_end\_date*) of a recurring payment. To enable this option, send a request to [merchantservices@skrill.com](mailto:merchantservices@skrill.com).

The following parameters are required:

**Table 7-7: Extend end date parameters**

Field Name	Description	Required?	Example value
trn_id	Your transaction ID.	Yes	500123
rec_end_date	The recurrent end date in <i>dd-mm-yyyy</i> format.	Yes	30-06-2017

Successful Response:

```
200 -> -> OK
```

The -> symbol represents a tab character.

Using escape sequences to represent special characters, the response is as follows:

```
200\t\tOK\n\n
```

### 7.1.7. Cancel a Skrill 1-Tap payment

Query parameter: *action= cancel\_od*

This action allows you to cancel a Skrill 1-Tap payment. The following parameter is required:

**Table 7-8: Cancel 1-Tap parameters**

Field Name	Description	Required?	Example value
trn_id	Your transaction ID.	Yes	500123

**Example successful cancel request response**

```
200 -> -> OK
```

The -> symbol represents a tab character.

Using escape sequences to represent special characters, the response is as follows:

```
200\t\tOK\n\n
```

### 7.1.8. View Skrill 1-Tap payment status

Query parameter: *action= status\_od*

This action allows you to check the status of a Skrill 1-Tap payment. The following parameter is required:

**Table 7-9: 1-Tap payment status parameters**

Field Name	Description	Required?	Example value
trn_id	Your transaction ID.	Yes	500123

If a transaction with the given ID is found, the response will contain following parameters:

- Status: *0* – active; *-1* – cancelled; *-2* – failed initial payment
- Last execution date in *dd-mm-yyyy* format. (– is used instead if no subsequent 1-Tap payments have been taken)

#### Example Response

```
200 ->    ->    OK
Status: -1  Last execution date: 08-01-2016
```

The *->* symbol represents a tab character. There are two spaces between the Status value and the word Last.

Using escape sequences to represent special characters, the response is as follows:

```
200\t\tOK\nStatus: -1  Last execution date: 08-01-2016\n
```

### 7.1.9. Get account balances

Query parameter: *action=balance*

This action allows you to get the current balances of your accounts. The following parameters are optional:

Field Name	Description	Required	Example Value
account_id	A single account id	No	131552

#### JSON Server Response

The server returns a JSON array containing objects of with the following structure:

Element	Type	Restrictions	Description	Example
account_id	number	64-bit positive signed integer	Account identifier	131552
curency	string	3 characters	ISO-4217 currency codes	USD
legalEntity	string	3 characters	Enum values “EEA” or “ROW”	ROW

balance	number	Floating-point numeric	Balance amount	12552.000000 254
---------	--------	------------------------	----------------	---------------------

**Notes:** If account\_id is specified, then only the given account information is returned, if it belongs to the merchant. Otherwise, an empty array is returned.

**legalEntity values:**

- EEA – European Economic Area
- ROW – Rest of the world

balance value contains total balance including reserves, not available balance, which excludes reserves.

**Example response**

```
[{
  "accountId": 131552,
  "currency": "USD",
  "legalEntity": "ROW",
  "balance": 12552.000000254
}, {
  "accountId": 131554,
  "currency": "EUR",
  "legalEntity": "EEA",
  "balance": -140.50}]
```

## 7.2. Error messages

The following error messages can be returned by the Merchant Query Interface:

**Table 7-10: MQI Error messages**

Error	Description	Reason for error
401	Unauthorised/ Cannot log in	Authentication is required and has failed or has not yet been provided.
402	Payment Required	Reserved for future use.
403	Forbidden	The request was a valid request, but the server is refusing to respond to it. For example, the provided credentials were successfully authenticated but do not grant the client permission to access the resource.
404	Not Found	The requested resource could not be found.
405	Method not Allowed	A request was made of a resource using a request method not supported. For example, using GET on a method which requires data to be presented via POST.

Example error HTTP response body (using escape sequences to represent special characters):

```
404\t\tIllegal parameter value: 17651712043534553\n
```

## 8. APPENDICES

### 8.1. ISO 4217 currencies

**Table 8-1: ISO 4217 Currencies accepted by Skrill**

<b>AED</b>	Utd. Arab Emir. Dirham	<b>MAD</b>	Moroccan Dirham
<b>AUD</b>	Australian Dollar *	<b>MYR</b>	Malaysian Ringgit
<b>BGN</b>	Bulgarian Leva	<b>NOK</b>	Norwegian Krone *
<b>BHD</b>	Bahraini Dinar	<b>NZD</b>	New Zealand Dollar *
<b>CAD</b>	Canadian Dollar *	<b>OMR</b>	Omani Rial
<b>CHF</b>	Swiss Franc *	<b>PLN</b>	Polish Zloty *
<b>CZK</b>	Czech Koruna ‡	<b>QAR</b>	Qatari Rial
<b>DKK</b>	Danish Krone *	<b>RON</b>	Romanian Leu New
<b>EUR</b>	Euro *	<b>RSD</b>	Serbian Dinar
<b>GBP</b>	British Pound *	<b>SAR</b>	Saudi Riyal
<b>HKD</b>	Hong Kong Dollar *	<b>SEK</b>	Swedish Krona *
<b>HRK</b>	Croatian Kuna	<b>SGD</b>	Singapore Dollar ‡
<b>HUF</b>	Hungarian Forint	<b>THB</b>	Thailand Baht
<b>ILS</b>	Israeli Shekel	<b>TND</b>	Tunisian Dinar
<b>INR</b>	Indian Rupee	<b>TRY</b>	New Turkish Lira ‡
<b>ISK</b>	Iceland Krona	<b>TWD</b>	Taiwan Dollar
<b>JOD</b>	Jordanian Dinar	<b>USD</b>	U.S. Dollar *
<b>JPY</b>	Japanese Yen *	<b>ZAR</b>	South-African Rand *
<b>KRW</b>	South-Korean Won	<b>COP</b>	Colombian Peso
<b>KWD</b>	Kuwaiti Dinar		

**Note:** \* = Supported by *Quick Checkout Payouts*.

‡ = Supported by *Quick Checkout Payouts*, excluding Mastercard.

## 8.2. Error Messages

**Table 8-2: Errors when validating parameters**

Error	Resolution
INVALID_OR_MISSING_ACTION	Invalid action or no action is provided
INVALID_*	Invalid parameter, * = name of parameter
INVALID_REC_PAYMENT_ID	Invalid recurring payment transaction ID
LOGIN_INVALID	Email address and/or password were not provided
MISSING_*	Missing parameter, * = name of parameter
PAYOUTS_NOT_ENABLED	Quick Checkout Payout feature is not enabled

**Table 8-3: Errors during log in**

Error	Resolution
CANNOT_LOGIN	Email address and/or API/MQI password are incorrect
PAYMENT_DENIED	Check in your account profile that the API is enabled and you are posting your requests from the IP address specified
NO_LOGIN_EXPLANATION	Merchant using non-existent email address

**Table 8-4: Errors when validating payment details**

Error	Resolution
ALREADY_EXECUTED	If you have requested that the value for frn_trn_id must be unique for each transfer, this error will be returned when you try to submit the same value for more than one transfer
BALANCE_NOT_ENOUGH	Sending amount exceeds account balance
CHECK_FOR_VERIFIED_EMAIL	Your account email address needs to be verified
COUNTRY_BLOCKED_FOR_PAYMENTS	The customer's country is blocked for outgoing balance payments
DISALLOWED_CROSS_REGION_PAYMENT	The <b>account_id</b> you have provided does not match the region of the recipient
DISALLOWED_RECIPIENT	You are not permitted to send money to the recipient. For example, gaming merchants are not permitted to send or receive payments to/from US based customers
EXECUTION_PENDING	If you resend a transfer request with the same session identifier before the 'transfer' request was processed, this error will be returned
INVALID_AMOUNT	Check amount format
INVALID_BNF_EMAIL	Check the format of the beneficiary email address
INVALID_CURRENCY	Check currency code

**Table 8-4: Errors when validating payment details (Continued)**

Error	Resolution
INVALID_FRN_TRN_ID	Check parameter length submitted
INVALID_NOTE	Check parameter length submitted
INVALID_SUBJECT	Check parameter length submitted
LL_NO_PAYMENT	Your account is locked for security reasons. Please contact us
MISSING_REGION_IDENTIFIER	A valid <i>account_id</i> value must be submitted to execute the transaction
SINGLE_TRN_LIMIT_VIOLATED	Maximum amount per transaction = EUR 10,000

**Table 8-5: Errors when making Skrill 1-Tap payment requests**

Error	Resolution
BALANCE_NOT_ENOUGH	The customer's account balance is insufficient
CARD_FAILED	The customer's credit or debit card failed
CUSTOMER_IS_LOCKED	The customer's account is locked for outgoing payments
MAX_AMOUNT_REACHED	The payment amount is greater than the maximum amount configured when 1-Tap payments were setup for this user.
MAX_REQ_REACHED	Too many failed Skrill 1-Tap payment requests to the API. For security reasons, only two failed attempts per user per 24 hours are allowed
ONDEMAND_CANCELLED	The customer has cancelled this Skrill 1-Tap payment
ONDEMAND_INVALID	The Skrill 1-Tap payment requested does not exist
RECIPIENT_LIMIT_EXCEEDED	The customer's account limits are not sufficient
REQUEST_FAILED	Generic response for transaction failing for any other reason

## 9. GLOSSARY

This section provides a description of key terms used in this guide.

Term	Explanation
Acquirer	An acquiring bank (or acquirer) is the bank or financial institution that processes credit and or debit card payments for a merchant.  Example: Barclays Merchant Service and European Merchant Services.
API	The API is a collection of tools that enables merchants to execute requests to the Skrill Payment Gateway. For example: to send money, make 1- payments, make refunds, check the status of transactions and download reports.
Back-end system	As opposed to a front-end system, a back-end system used internally by Skrill or within the merchant's business. Skrill merchants can also use payment information returned from the Skrill Payment Platform on their own back-end systems, such their customer order management system.
Batch	A group of approved credit card transactions, accumulated during one business day (weekends and official/bank holidays excluded).
Browser	Application that enables a customer or merchant to access web pages. Examples include: Internet Explorer, Google Chrome and Mozilla Firefox.
Cancel	Request to cancels a transaction. This is only possible before you have captured the payment or until the preauthorisation expires. It can be for a partial amount.
Customer ID	On the <b>Skrill Payment Platform</b> , a merchant may be configured with multiple customer IDs set up for different channels.  (Not to be confused with the customerid field.)
Customer services team	Skrill team responsible for end-customer support queries. Also referred to as the <b>Merchant Services team</b> .
Integration	Process undertaken by merchants to ensure that their website or shopping cart can connect to and communicate with Skrill's payment processing systems.
ISO country codes	3-digit country code of the International Standards organisation (ISO) that identifies the country. For example, GBR for United Kingdom. ISO country codes also exist in a 2-digit format.
ISO currency codes	3-digit currency code of the International Standards Organisation (ISO) that identifies the currency. For example, GBP for British Pound.
MD5	A widely used hash algorithm, which can be used for securely encrypting information sent over the internet. MD5 produces a 128-bit (16-byte) hash value. The purpose of the field is to ensure the integrity of the data posted back to the merchants' server.
Merchant	Skrill customer (legal or natural person) using their Skrill solution to receive payments for products or services they provide.

Term	Explanation
Merchant ID (MID)	A merchant identifier, provided by the <b>Acquirer</b> , used to uniquely identify a merchant within the banking network when a transaction is processed.
Merchant Services team	Skrill team responsible for providing technical and service support to merchants.
My Account	Customer account administration portal that enables viewing of transactions and transferring funds.
On-boarding process	Process of signing up and verifying a merchant. This involves a number of teams in Skrill, including sales and risk and compliance. On the payment processing side, this process is coordinated by a dedicated on-boarding team.
Payment	Unique financial record on the system. A payment may consist of multiple <b>Transactions</b> .
Payment processing platform	System used for the processing of eCommerce transactions.
Pending transaction	A transaction in which the payment system is waiting for a confirmation, an input or customer action.
Refund	Option to pay money back to a customer, which can be done using the <b>API</b> .  The refund has to be referenced to the original payment and can only be up to that amount. Skrill enables partial or full amount refunds.
Real-time	An event that occurs instantly or within a short period, such as seconds or minutes. For a real-time transaction, the customer, merchant or Skrill receive a response to the transaction request while the customer is still online.
Skrill Digital Wallet	The Skrill Digital Wallet allows customers to link cards and pay directly from their wallet account using cards or bank transfer. Up to 4 payment cards and 10 bank accounts can be linked to a wallet account.
Skrill Quick Checkout	Skrill product, related to the <b>Skrill Digital Wallet</b> , which enables customers to bypass the Skrill registration details page and simply confirm and pay. Quick Checkout uses the eCommerce platform for processing payments.
Skrill Payment Platform	Skrill's system for the processing of eCommerce payments.
Transaction	Each financial interaction with the <b>Skrill Payment Platform</b> is referred to as a transaction. Transactions are linked to <b>Payments</b> .
Transaction ID	Unique ID assigned to a transaction by the <b>Skrill Payment Platform</b> .
Transaction status	Each transaction on the <b>Skrill Payment Platform</b> is given a status. This includes: processed, pending, temporary, scheduled, cancelled, failed, chargeback and successful.



---

## INDEX

---

### A

account history, viewing 27

reposting an execution request 11

### C

currencies accepted by Skrill 32

### E

error messages 33

### M

MD5 signature 23

Merchant Query Interface (MQI) 24

error messages 31

MQI and API

password 5

### Q

Quick Checkout payouts, see transfer (payouts)

### R

recurring payment

canceling 28

extending end date 29

status, viewing 28

refund

error messages 20

execution 21

preparation 18

status report 22

### S

security 4

Skrill 1-Tap payment

canceling 29

viewing status 30

### T

transaction status

reposting 25

viewing 25

transfer (payouts) 12

example request and response 15

executing 14

prepare request 12

response 14

transfer (send money) 7

error LOCK\_LEVEL\_9 9

example request and response 10

executing 9

prepare request 7